Automated Group-Contribution Predictions of Properties Within the DIPPR 801 Database

R. Jeremy Rowley, W. Vincent Wilding,* John L. Oscarson, and Richard L. Rowley

Department of Chemical Engineering, 350CB, Brigham Young University,

Provo, UT 84602

* Corresponding author. E-mail: wildingv@byu.edu Fax: 801-378-7799

Abstract:

Two hallmarks of the DIPPR Project 801 Database are (1) a complete set (whenever possible) of 44 properties for each compound included in the database and (2) a thorough evaluation of the accuracy of the property values entered. Completeness requires that predictive methods be used when experimental data are not available. Evaluation of predicted properties further requires an understanding of the probable accuracy of the chosen method for the compound. Combination of automated prediction methods with a large experimental database makes it possible to compare predictions to large quantities of data and analyze prediction methods for broad classes of compounds or individual compound families.

Group contribution methods are common and valuable prediction techniques. These methods traditionally require manual splitting of each compound into groups. We report here an automated string parser that works in conjunction with a group assignment algorithm and a property calculator. Together these algorithms constitute a powerful, automated group-contribution prediction package.

The parser utilizes SMILES[1] (Simplified Molecular Input Line Entry System) formulas, which are available for each component in the DIPPR database. The parser makes several passes through the SMILES string, populating two arrays: one that shows group position, type of group, position of all attached neighboring groups, and types of neighboring groups; and another that contains ring type and ring closure information. The group assignment algorithm then utilizes these arrays to identify matches with stored group definitions and counts the number of each identified group. The property calculator then calculates the property using the desired group-contribution method.

Introduction

For twenty years DIPPR Project 801, Evaluated Process Design Data, has been dedicated to developing, organizing and making available to project sponsors a complete, critically-evaluated compilation of thermophysical properties of industrial important chemicals. The database currently contains over 1750 chemicals. New compounds are continually added to the database and the compounds in the database are periodically reviewed for accuracy. Two of the most significant aspects of this database are completeness and critical evaluation. When a compound is added to the database values for all of the database's 44 properties (29 property constants and 15 temperature-dependent properties) are included. This completeness requires that predictive methods be used when experimental data are not available. Evaluation of predicted properties further requires an understanding of the probable accuracy of the chosen method for the compound of interest.

In the past few years DIPPR Project 801 has implemented state-of-the-art computer tools to improve usability and quality of the database. As part of this implementation, the database was converted from an ASCII flat file format to a relational database format. This change facilitates data retrieval and quality control. Combining the relational database with automated prediction methods makes it possible to efficiently evaluate predictive methods.

Throughout the course of DIPPR Project 801 prediction methods have been evaluated and selected to be used when experimental data are unavailable. As new methods are promulgated these are compared to methods currently in use and if shown to be superior they are incorporated into the project's regular procedures. A variety of prediction methods are available for most of the properties in the DIPPR database. Prediction methods are classified as *primary*, *secondary*, or *tertiary* for a given property depending upon their

reliability. Determining which prediction method is best suited for a given compound (i.e. which method gives the most accurate prediction) is not always an easy task. Most prediction methods are group-contribution methods. These methods determine a property value by summing the contributions to the value contributed by each of the compound's functional groups. To apply a group contribution method, the compound is split into the composing groups and then the property is calculated from the combination of the group-specific contributions. Manually splitting compounds into their constituent groups is slow, tedious, and often encumbered by ambiguities on group designations. Also, many methods lack contributions for certain groups which limits the applicability of these methods.

By automating the tedious splitting step of the prediction method and by coupling this capability with the evaluated DIPPR database, prediction methods can be efficiently tested and compared, and new methods can be developed. This will significantly enhance the DIPPR database by improving the property predictions included in the database.

Results

The need for automation has led to the development of a prediction package that can be used with DIPPR's DIADEM program. (DIADEM is a program for accessing and displaying the DIPPR database and is written in Visual Basic®.) This allows predictions to be made using the DIPPR database. The prediction package utilizes the SMILES[1] (Simplified Molecular Input Line Entry System) formulas to determine group contributions. The groups found by the prediction package are then used in conjunction with a property calculator to predict a property value.

SMILES Parser

The SMILES parser used by DIADEM was developed to parse SMILES formulas into different chemical groups. These groups are based on the Domalski-Hearing[2] (DH) group definitions. The Domalski-Hearing method was chosen because of its generality. Groups for most other methods can be obtained from DH groups with only slight modification. The DH method provides a wide set of nearest-neighbor groups that can be readily identified in a compound. The parser makes two passes through a formula. The first pass breaks the SMILES formula into its unique DH groups. The second pass detects structural combinations such as individual and fused rings.

Group Routine

The first parser pass through the SMILES formula breaks the formula up into groups and stores them in an array. This array contains information regarding the group's ID, position within the formula, and groups to which the group under consideration is attached. The connections to the group are based on 1) any branching attachments made through use of parentheses, 2) any direct attachments (outside of parentheses), and 3) any ring openings/closings that occur at the group location. Direct attachments can be either before or after the group. For example, the SMILES formula NCN indicates two N direct attachments to the C group. In turn each N has one C direct attachment.

Parentheses are used in a SMILES formula to show branching. Parenthetical attachments are determined by finding the first group listed after the opening parenthesis. After the parentheses are open, additional attachments are not found until the parentheses are closed. Parentheses attachments are searched for after the group. This process

continues until a direct attachment is found. Once a direct attachment is located no more parentheses attachments are possible. As an example consider the SMILES formula C(B)(B)NC(B)(B) where B can represent any additional groups in the chain including additional sets of parentheses. In this case each C group would have one direct attachment (N) and two parentheses attachments.

Connections made through ring openings are found by searching the string for the ring closure and adding the first direct attachment found previous to the closure. The ring-opening group is then stored in an array that can be referenced later when the ring closes. For benzene (c1ccccc1), this procedure allows each c to be attached to two other c groups.

A difficulty encountered when developing the parser was accounting for double and triple bonds. Often the bonds may be separated by several parentheses. To find any double bonds that might exist, each direct attachment and parentheses set is evaluated for any = or # falling between the attachment and the group. For ring connections, if both the ring closure and opening has a =, as in C=1C…CC=1, only one is counted as a real double bond, splitting the group into a C= not a C==. Figure 1 shows 1-methylnaphthalene and Table 1 displays the groups found in the SMILES formula, their position, and the groups attached to the specified group. (The smiles formula for 1-methylnaphthalene is c1(C)cccc2ccccc21.)

Ring Identification

The second pass with the parser program searches the SMILES formula for different types of ring structures. Because of the variety of ring structure ID's in various group methods, the ring structures found by DIADEM are based on the smallest rings present. By breaking the rings up and storing their interconnection information, different methods to

rebuild and use the rings can be employed by different group methods without reparsing. Only the smallest ring structures found in a compound are stored by DIADEM. Naphthalene, for example, is stored as a combination of two six-member rings. Adamantane is broken into three intertwined six-member rings. Such a search does have a few problems. Some ring structures can be broken up in a variety of ways. Norbornane (see Figure 2) can be viewed as either two five-member rings or a six-member ring and a five-member ring.

Since prediction methods that lack a specific norbornane group contribution do not specify what should be done in this situation, the ring information stored in the *ring* array for these compounds is dependent on the how the SMILES formula is written.

The first step in the ring analysis breaks the ring down into individual components. These components include information on the groups that are included in the ring, their position within the SMILES formula, where the ring opens and closes in the SMILES formula, and how many members each ring has. This information can be accessed by other parts of the program. Groups are added to the ring much like groups were found in the previous pass. To determine which groups are part of the ring, the parser begins at the ring opening and moves through the SMILES formula examining each letter between the ring opening and closing to determine whether or not the selected letter is part of the ring. This decision is based on where the group falls within the ring formula, in a set of parentheses, or within another ring's opening or closing. If the parser determines that the selection meets the necessary criteria, the group ID and position are added as ring members. Once the group analysis is complete the routine moves on to the next group in the SMILES formula, looping until all of the ring's members have been found.

An open parenthesis in a SMILES formula means that a compound has some branching. Branches in rings are handled in DIADEM by a special subroutine that searches the SMILES formula for the matching closing parenthesis. If the matching closing parenthesis occurs after the ring's closing group, the SMILES formula ignores the set of parentheses and continues as normal. If instead the closing parenthesis is located before the ring's close, the parentheses are passed over and the parser moves to the next group found after the parentheses' end. This means that if the parser finds the SMILES structure CN(CCC)*y*C within a ring, it will add the first N then skip everything inside the parentheses and add *y* as the next group in the ring. Once the parser adds the N group in the SMILES formula CN(C(CC)CC(C)C((CC)C))*y*C, the parser will skip to the *y* group (the group after the corresponding closing parenthesis) to resume parsing.

Another special situation is when the parser encounters multiple ring structures. If the ring finder finds a position within the ring where a second ring opening occurs then a new subroutine is called to count how many groups are held in common between the two rings. Groups common to both rings are ones that are included in both ring structures. Knowing how many groups are common between the two ring structures makes splitting the multi-ring into its smaller ring components possible.

The first group in common is the group that opens the second ring (called R2). R2 is added to the common count, then the parser loops through the SMILES formula looking for additional groups held in common. Groups are considered to be in common if the group is not set off by parentheses from one of the rings and is not an additional ring opening/ closing group. If a set of parentheses opens or a third ring opens, the parser stops counting groups in common until the parentheses/additional ring closes. In addition, in rings that have a link

group (rings that open inside a set of parentheses that do not contain the ring's end group) the link group is considered to be in common. If the group is held in common between R2 and the primary ring (R1), then the group is added to the common count. Several examples of groups held in common and groups not in common are provided in Table 2. In each of the examples the letter D is used to represent groups in common.

After the common counter is finished, the ring structure analysis can be completed. If less than three groups are found to be in common and no additional rings are opened, the common groups are added to the *ring* array. However, if the common count is three or more, a special subroutine is called to finish the parsing. How the parser finishes depends on the SMILES structure.

While parsing the ring, the ring's subroutine stores information on whether or not each member of the ring has groups which are not part of any ring attached to it. With this information, the ring finder is able to keep track of possible *ortho* and *meta* combinations.

Translating the Information

After the SMILES formula is analyzed for groups and rings, the information found by the parser is translated into different class objects within Visual Basic®. These class objects are then used throughout the prediction package for determining prediction values. Using class modules allows new prediction methods to be added easily to the property calculator.

Each prediction method is fitted with specific code that allows the property to be calculated from the group class modules and required parameters. Often this code requires recombining groups to form additional groups. For example, many methods require the

group COO. This group can be made by combining the CO group with an attached O group. For some prediction methods, the recombined groups can be several different groups. A COOCO group also contains the COO and CO groups. In these cases a group priority system is setup; certain groups take precedence over others. When these groups are found, groups of lower priority are ignored in favor of a higher priority group. In the previous example, COOCO might be a priority 1 group, COO a priority 2 group and CO a priority 3 group. If the priority 1 group is found, the two COs in the priority 1 group will not be counted as a priority 2 or priority 3 group.

To enable rapid comparisons, the user interface to the prediction methods consists of a form that allows calculation of several families of compounds at once. The form shown in Figure 3 allows users to make predictions of families selected from a list.

The form allows users to readily view predictions and errors for each of the chemicals in the family as well as an overall error and deviation. This allows researchers to evaluate different prediction methods in a matter of minutes. This form can further restrict predictions based on the accuracy of the stored database value. If the value stored in the database is not known to be accurate, the prediction's deviation from the stored value isn't as useful in evaluating the prediction method.

Conclusions

The addition of the SMILES parser and prediction package to the DIPPR database has greatly enhanced the efficiency with which predictions can be performed and evaluated and will be a useful tool for the development of new prediction techniques.

Acknowledgment

We gratefully acknowledge the support of this work by the DIPPR 801 Project.

Literature Cited

(1) D. Weininger, “SMILES, a Chemical Language and Information System,” *J. Chem. Infor. Comp. Sci.*, 28, 31, 1988.

(2) E. S. Domalski and E. D. Hearing, “Estimation of the Thermodynamic Properties of C-H-N-O-S-Halogen Compounds at 298.15 K,” *J. Phys. Chem. Ref. Data*, 22(4), 805, 1993.

Table 1. Groups and Their Attachments for 1-methylnaphthalene

Group	Position	Attachments (Position)
C=	1	C=(5), C=(15), C=(23)
C=	5	C=(1), C=(7), C=(18)
C=	7	C=(5), C=(9)
C=	9	C=(7), C(11), C=(13)
C	11	C=(9)
C=	13	C=(9), C=(15)
C=	15	C=(1), C=(13)
C=	18	C=(9), C=(20)
C=	20	C=(18), C=(21)
C=	21	C=(20), C=(23)
C=	23	C=(1), C=(21)

Table 2. Examples of Groups in Common

Structure	Common Count	Comments
C1CD2DDDD2CCCC1	5	The common counter starts when a second ring is opened and stops when the ring closes.
C1CD2D(CCC)DD2CC1	2	The common counter function ceases until the end of the parentheses.
D12D(cccc1)ccccc2	4	The common counter stops counting at the parentheses. Since the ring ends before the close of the parentheses, no additional common groups are found after the parentheses close.
C1CD2D3CCC3DD2C1	4	Once an additional ring is opened the common counter stops until the ring is closed.

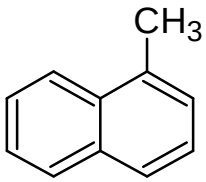

Figure 1. 1-Methylnaphthalene

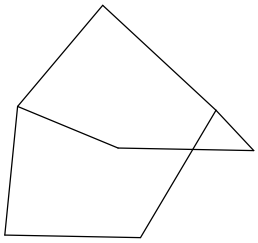

Figure 2. Structure of Norbornane

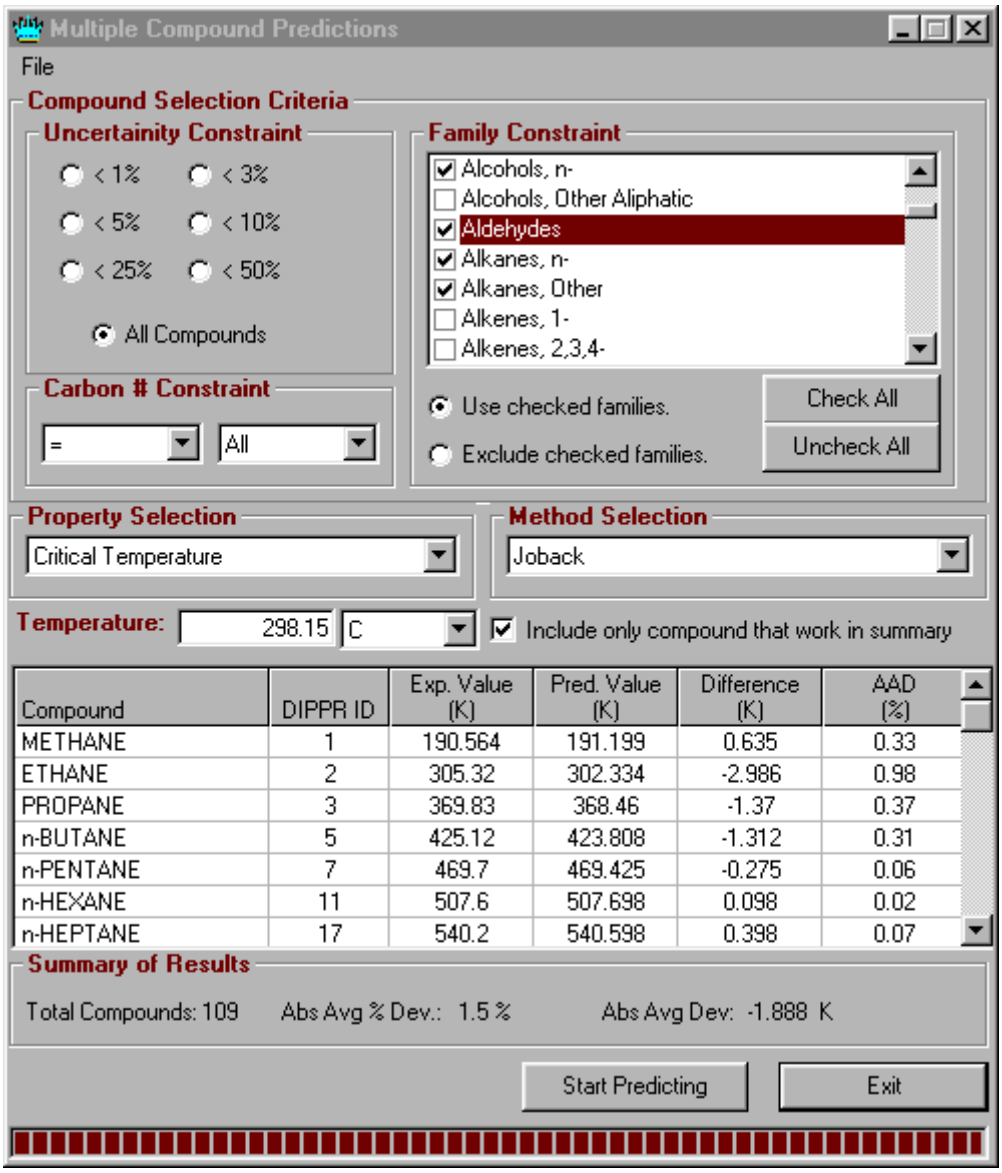

Figure 3. Family Predictions Form